

# 7 - Indirect Communication

---

## 17S1

---

**Q5 (a): [5 marks] What are the differences between the message queue paradigm and the publish/subscribe paradigm? Give an example application where the message queue paradigm would be the more appropriate choice over publish/subscribe and say why.**

The publisher will push messages to a Topic, and the Topic will instantly push the message to all the subscribers. This is what differentiates the Pub/Sub model from traditional message brokers, where **a message queue will batch individual messages until a user or service requests these messages and retrieves them.**

Message queue is more suitable in chat apps. Messages to an offline user can be stored in the queue. When the user is online, only the messages for the user will be retrieved.

## 16S2

---

**Q5: (a) [5 marks] Consider a “music popularity” application that runs on mobile devices. Each user’s mobile device reports the name of the song that is currently playing on the device, to a central server. The central server ranks the songs in terms of popularity across all devices and makes the results known back to all users, so that the most popular songs can be seen by everyone. Choose either tuple space, message queue or publish/subscribe paradigm and explain how the application could be implemented using your chosen paradigm. Draw an architectural diagram to aid your explanation.**

Tuple space: The list of most popular songs can be regarded as an abstract form of distributed shared memory or database among all the users. The list is dynamically updated according to online users who playing music.

**(b) [5 marks] Consider implementing a distributed shared memory middleware. Describe what kind of protocol and messages that you would use, and why. Discuss the operating system support that your implementation would require.**

Use Java RMI to exchange messages. Need to have concurrency control. JVM supports almost all OS.

## 15S2

---

**Q4: (a) [3 marks] Describe distributed shared memory, including what it is used for, how it is used by applications, and how it works. L7 P19**

**(b) Message Queue vs. Tuple Space**

- Message Queue
  - Easier to implement
  - Good scalability
- Tuple Space
  - Describe parallel algorithms without reference to any specific computer architecture
  - Supports a shared-memory store

**(e) [4 marks] Describe the Publish/Subscribe paradigm. Use a diagram and include a description of the API used in a Publish/Subscribe system.**

L7 P13-17

## Other

---

**Explain the following aspects of a publish subscribe system: P118**

- Event
  - A change of state that is of interest.
- Notification
  - Information regarding an event that is sent to a subscriber.
- Subscriber
  - Express interest in particular events and get notified when they occur.
- Publisher
  - Publish/send structured events to an event service.

**What are the advantages of message queues compared to traditional sending and receiving? What problems can be caused by using message queue system in a distributed system?**

Advantages:

- Uncoupling P114
  - Traditional sending and receiving require high coupling.
  - By using message queues, time uncoupling can be achieved so that sender and receiver can have independent life times. They do not need to exist simultaneously.
- Asynchronous
  - Non-essential business logic runs asynchronously to speed up response.
- Peak clipping
  - Message queue can serve as a buffer for the server. Especially when there are a huge amount of requests. The messages can be placed in the queue until the server is able to process them.

#### Problems:

- The system becomes more complex. Both the server and the message queue need to be maintained. The system's availability is reduced since it will fail if the message queue fails.
- Need to deal with consistency.
- Need to ensure reliable transmission of messages.
- Need to ensure that messages are not repeatedly consumed.